CLAIMS

1. A computer-implemented method for filtering an image including a plurality of pixels, the method comprising:

receiving a forward kernel centered at a first pixel in the image, the forward kernel assigning forward weights to pixels in a neighborhood surrounding the first pixel;

specifying a backward kernel centered at a second pixel within the neighborhood surrounding the first pixel based on a local attribute of the image at the second pixel, the backward kernel assigning backward weights to pixels in a neighborhood surrounding the second pixel;

determining a convolution weight of the second pixel based on the backward kernel and the forward kernel; and

using the convolution weight and a pixel value of the second pixel to generate a new value of the first pixel.

2. The method of claim 1, wherein determining the convolution weight of the second pixel includes:

determining a forward weight assigned to the second pixel by the forward kernel;

determining a backward weight assigned to the first pixel by the backward kernel; and

using the forward weight and the backward weight to determine the convolution weight of the second pixel.

3. The method of claim 2, wherein:

determining the convolution weight of the second pixel includes multiplying the forward weight and the backward weight.

4. The method of claim 2, wherein:

determining the convolution weight of the second pixel includes setting the value of the convolution weight to the smaller of the forward weight and the backward weight.

5. The method of claim 1, wherein:

determining the convolution weight of the second pixel includes specifying a substantially zero value for the convolution weight of the second pixel if the backward kernel assigns a substantially zero backward weight to the first pixel.

6. The method of claim 1, wherein:

determining the convolution weight of the second pixel includes specifying a substantially non-zero value for the convolution weight of the second pixel if the backward kernel assigns a substantially non-zero backward weight to the first pixel.

7. The method of claim 6, wherein the non-zero value for the convolution weight of the second pixel is a predetermined value.

8. The method of claim 6, wherein the non-zero value for the convolution weight of the second pixel is a forward weight assigned to the second pixel by the forward kernel.

9. The method of claim 1, wherein the local attribute of the image at the second pixel is a depth value corresponding to a distance of an object represented by the second pixel relative to a focal distance.

10. The method of claim 9, further comprising:

receiving user input specifying a depth map assigning a depth value to each pixel in the image.

11. The method of claim 1, wherein the local attribute of the image at the second pixel is a luminance value.

12. The method of claim 1, wherein:

the received forward kernel is operable to blur the image at the first pixel.

13. The method of claim 12, wherein:

the specified backward kernel is operable to blur the image at the second pixel.

14. The method of claim 1, wherein:

the received forward kernel is operable to sharpen the image at the first pixel.

15. The method of claim 1, wherein:

receiving the forward kernel centered at the first pixel includes receiving an array of forward weights, each forward weight in the array being assigned to a pixel in the neighborhood surrounding the first pixel.

16. The method of claim 1, wherein:

receiving the forward kernel centered at the first pixel includes receiving a kernel function having a kernel location at the first pixel and specifying a forward weight to each pixel in the neighborhood surrounding the first pixel based on a distance between the kernel location and the pixel in the neighborhood.

17. The method of claim 16, wherein:

the kernel function depends on a kernel radius; and

specifying the backward kernel centered at the second pixel includes determining a kernel radius based on the local attribute of the image at the second pixel and specifying the backward kernel by the kernel function with the determined kernel radius and a kernel location at the second pixel.

18. The method of claim 1, further comprising:

specifying one or more further backward kernels, each of the further backward kernels being centered at a corresponding further pixel within the neighborhood surrounding the first pixel and assigning backward weights to pixels in a neighborhood surrounding the corresponding further pixel, each of the further backward kernels being based on a local attribute of the image at the corresponding further pixel;

determining a convolution weight of each further pixel based on the corresponding backward kernel and the forward kernel; and

using the convolution weight and a pixel value of each further pixel to generate the new value of the first pixel.

19. A computer-implemented method for depth of field filtering an image including a plurality of pixels, the method comprising:

specifying a plurality of forward kernels, each of the forward kernels being centered at a corresponding center pixel in the plurality of pixels and being based on a depth value assigned to the corresponding center pixel;

for each of the forward kernels, determining convolution weights of neighborhood pixels within a neighborhood surrounding the center pixel of the forward kernel, each neighborhood pixel's convolution weight being determined based on the forward kernel and a backward kernel that is centered at the neighborhood pixel and is based on a depth value assigned to the neighborhood pixel; and

blurring the image at each center pixel of the forward kernels using the convolution weights and pixel values of the neighborhood pixels within the neighborhood surrounding the center pixel.

20. A software product, tangibly embodied in a machine-readable medium, for filtering an image including a plurality of pixels, the software product comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

receiving a forward kernel centered at a first pixel in the image, the forward kernel assigning forward weights to pixels in a neighborhood surrounding the first pixel;

specifying a backward kernel centered at a second pixel within the neighborhood surrounding the first pixel based on a local attribute of the image at the second pixel, the backward kernel assigning backward weights to pixels in a neighborhood surrounding the second pixel;

determining a convolution weight of the second pixel based on the backward kernel and the forward kernel; and

using the convolution weight and a pixel value of the second pixel to generate a new value of the first pixel.

21. The software product of claim 20, wherein determining the convolution weight of the second pixel includes:

determining a forward weight assigned to the second pixel by the forward kernel;

determining a backward weight assigned to the first pixel by the backward kernel; and

using the forward weight and the backward weight to determine the convolution weight of the second pixel.

22. The software product of claim 21, wherein:

determining the convolution weight of the second pixel includes multiplying the forward weight and the backward weight.

23. The software product of claim 21, wherein:

determining the convolution weight of the second pixel includes setting the value of the convolution weight to the smaller of the forward weight and the backward weight.

24. The software product of claim 20, wherein:

determining the convolution weight of the second pixel includes specifying a substantially zero value for the convolution weight of the second pixel if the backward kernel assigns a substantially zero backward weight to the first pixel.

25. The software product of claim 20, wherein:

determining the convolution weight of the second pixel includes specifying a substantially non-zero value for the convolution weight of the second pixel if the backward kernel assigns a substantially non-zero backward weight to the first pixel.

26. The software product of claim 25, wherein the non-zero value for the convolution weight of the second pixel is a predetermined value.

27. The software product of claim 25, wherein the non-zero value for the convolution weight of the second pixel is a forward weight assigned to the second pixel by the forward kernel.

28. The software product of claim 20, wherein the local attribute of the image at the second pixel is a depth value corresponding to a distance of an object represented by the second pixel relative to a focal distance.

29. The software product of claim 28, further comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

receiving user input specifying a depth map assigning a depth value to each pixel in the image.

30. The software product of claim 20, wherein the local attribute of the image at the second pixel is a luminance value.

31. The software product of claim 20, wherein:

the received forward kernel is operable to blur the image at the first pixel.

32. The software product of claim 31, wherein:

the specified backward kernel is operable to blur the image at the second pixel.

33. The software product of claim 20, wherein:

the received forward kernel is operable to sharpen the image at the first pixel.

34. The software product of claim 20, wherein:

receiving the forward kernel centered at the first pixel includes receiving an array of forward weights, each forward weight in the array being assigned to a pixel in the neighborhood surrounding the first pixel.

35. The software product of claim 20, wherein:

receiving the forward kernel centered at the first pixel includes receiving a kernel function having a kernel location at the first pixel and specifying a forward weight to each pixel in the neighborhood surrounding the first pixel based on a distance between the kernel location and the pixel in the neighborhood.

36. The software product of claim 35, wherein:

the kernel function depends on a kernel radius; and

specifying the backward kernel centered at the second pixel includes determining a kernel radius based on the local attribute of the image at the second pixel and specifying the backward kernel by the kernel function with the determined kernel radius and a kernel location at the second pixel.

37. The software product of claim 20, further comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

specifying one or more further backward kernels, each of the further backward kernels being centered at a corresponding further pixel within the neighborhood surrounding the first pixel and assigning backward weights to pixels in a neighborhood surrounding the corresponding further pixel, each of the further backward kernels being based on a local attribute of the image at the corresponding further pixel;

determining a convolution weight of each further pixel based on the corresponding backward kernel and the forward kernel; and

using the convolution weight and a pixel value of each further pixel to generate the new value of the first pixel.

38. A software product, tangibly embodied in a machine-readable medium, for depth of field filtering an image including a plurality of pixels, the software product comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

specifying a plurality of forward kernels, each of the forward kernels being centered at a corresponding center pixel in the plurality of pixels and being based on a depth value assigned to the corresponding center pixel;

for each of the forward kernels, determining convolution weights of neighborhood pixels within a neighborhood surrounding the center pixel of the forward kernel, each neighborhood pixel's convolution weight being determined based on the forward kernel and a backward kernel that is centered at the neighborhood pixel and is based on a depth value assigned to the neighborhood pixel; and

blurring the image at each center pixel of the forward kernels using the convolution weights and pixel values of the neighborhood pixels within the neighborhood surrounding the center pixel.